Función de la evaluación de las particiones en bases de datos distribuidas orientadas a objetos.

Resumen

Las bases de datos orientadas a objetos se utilizan cada vez con más frecuencia. Estas bases datos heredan muchos de los conceptos desarrollados para las bases de datos relacionales. De modo similar, las bases de datos distribuidas relacionales tienen estrecha conexión con bases de datos distribuidas orientadas a objetos. En este artículo se presenta el desarrollo de una nueva función llamada Partition Evaluator for Object-Oriented Databases que permite evaluar las particiones "verticales" en bases de datos distribuidas orientaras a objetos. Se demuestra que esta función es la generalización del así llamado Partition Evaluator desarrollado para bases de datos distribuidas realcionales.

1. INTRODUCCIÓN

Las bases de datos distribuidas relacionales (BDDR) fueron investigadas durante muchos años. Sin embargo, el interés por su desarrollo creció conforme las compañías se expandieron geográficamente y se implementaron las redes de computadoras. Son presentadas como una colección de múltiples y lógicamente relacionadas bases de datos (BD). Deben ser diseñadas tomando en cuenta todos los aspectos de la BD centralizada y además incluir los aspectos relacionados con la fragmentación, la replicación y la localización para mejorar su desempeño [15].

Uno de los aspectos ampliamente investigados en BDDR es la fragmentación. El enfoque relacional distingue dos tipos de fragmentación: horizontal, vertical y también la fragmentación híbrida que está formada por los dos tipos de fragmentación mencionados anteriormente. Cada una de ellas tiene sus propios criterios para determinar el mejor esquema de partición para el sistema relacional ([2, 3, 4, 12, 13, 14]).

Por otro lado, aunque se han hecho un número considerable de investigaciones con respecto a las bases de datos orientaras a objetos (BDOO), muy pocas se relacionan con la distribución de los objetos y específicamente con la fragmentación de éstos. Esta última es estudiada por Ezeife y Barker [5] quienes mencionan que sus beneficios deberían ser también reconocidos en BDOO. Aunque existe una diferencia significante entre las BD relacionales y BDOO, debe ser claro que el enfoque relacional puede ser visto (ignorando por la simplicidad la existencia de los métodos) como un caso especial de sistemas orientados a objetos sin la jerarquía de clases y atributos complejos (contienen la referencia a otros objetos). Por consiguiente, muchos de los algoritmos desarrollados para BDDR pueden ser generalizados y aplicados para BDOO, como lo es el evaluator de particiones (PEOO, partition evaluator for object-oriented databases) que puede ser aplicado en la búsqueda exhaustiva para evaluar cada una de las combinaciones de la fragmentación vertical de los objetos o para evaluar la "bondad" de algoritmos de fragmentación. Este evaluador de particiones es la generalización del evaluador de particiones (PE, parition evaluator) desarrollado para las bases de datos relacionales y presentado por Chakravarthy et al. [1].

El presente artículo menciona los aspectos e investigaciones hechas con respecto a la fragmentación tanto en bases de datos relacionales como orientadas a objetos. Además, en forma breve se presenta la motivación para el desarrollo de PEOO y se especifica el desarrollo de la función objetiva, presentando las notaciones usadas y los pasos para obtener la fórmula para el costo local y el costo remoto.

2. ASPECTOS DE LA FRAGMENTACION EN BASES DE DATOS RELACIONALES Y BASES DE DATOS ORIENTADAS A OBJETOS

La fragmentación vertical permite al diseñador agrupar los atributos de cada relación en registros más pequeños para mejorar la eficiencia de las transacciones en bases de datos. Sin embargo, el número de las particiones posibles con m atributos presentes es igual a m número de Bell; para bases de datos grandes este número es igual a [m.sup.m] [13]. Para hacer este número más pequeño, en la mayoría de las investigaciones se usa un enfoque heurístico. Una de las suposiciones más usadas en este enfoque es que cada fragmento debe estar muy cerca de los requerimientos de la transacción que lo ocupa [13]. Esto significa, que en el caso ideal el fragmento puede tener sólo los atributos que la transacción local necesita y ésta no necesita accesar ninguno de ellos de otro nodo remoto, lo cual no es real. La mejor forma de partir los atributos para una transacción no es necesariamente la mejor forma para otra transacción. La meta factible es maximizar el desempeño de todas las transacciones tomándolas como un solo grupo, pues, el diseñador debe asegurar el acceso mínimo a los atributos remotos para todas las transacciones ejecutadas en cada uno de los sitios.

El seguimiento de estos conceptos se puede ver en las investigaciones presentadas sobre la fragmentación vertical en BDDR. Actualmente existe un número considerable de algoritmos para la fragmentación vertical en BD relacionales (Hoffer y Severance [8], Hammer y Niamir [7], Navathe et al. [13], Cornell y Yu [4], Navathe y Ra [14], Chu y Ieong [3]). En la mayoría de estos algoritmos se usa la matriz de afinidad de los atributos (AAM, atribute affinty matrix). Sin embargo, en este artículo se enfatiza más el enfoque presentado por Chakravarthy et al. [1] que defiende el uso de la matriz del uso de atributos (AUM, attribute usage matrix) en lugar de AAM. Chakravarthy et al. opinan que AAM solamente señala la "afinidad" entre las parejas de los atributos y las relaciones entre más de dos de ellos no se pueden medir en esta matriz. Basándose en la AUM ellos establecen la función que puede medir la "bondad" de un esquema de fragmentación. Esta función se puede usar para el proceso de fragmentación (búsqueda exhaustiva) o puede ser aplicada después de usar algún otro algoritmo de fragmentación.

Aunque los conceptos de fragmentación desarrollados para BDDR son ampliamente aceptados, los algoritmos de la fragmentación presentados para BDOO, en general, no se basan en ellos. Algunos autores presentan algoritmos desarrollados e implementados para fragmentar BDOO (Ezeiffe y Barker [5], Gruber y Amsleg [6]), estableciendo nuevos conceptos como afinidad entre objetos, fragmentos derivados, ligas de jerarquía, y otros. Otros autores analizan en forma general la posibilidad de la fragmentación sin ninguna implementación (Karlapalem y Li [10], Karlapalem et al. [11]. Esta falta de unificación de la visión con respecto a lo que es la fragmentación en BDOO hace difícil la motivación de los investigadores para profundizar en esta área.

3. MOTIVACIÓN PARA EL DESARROLLO DE LA NUEVA HERRAMIENTA PARA LA FRAGMENTACION VERTICAL EN BDOO

En el enfoque relacional los algoritmos de fragmentación vertical distribuyen los atributos. Sin embargo, en el enfoque orientado a objetos el esquema conceptual puede ser mucho más complejo, lejos de factores usados en el enfoque relacional (frecuencias de las transacciones, uso de los atributos, tipo de predicado, y otros). Adicionalmente deben ser considerados algunos aspectos nuevos, como: métodos, estructura jerárquica y los atributos complejos.

Además, la AUM usada en la fragmentación vertical de los trabajos previos en BDDR, solamente considera el caso cuando las frecuencias de acceso de los atributos por una transacción específica son iguales para todos los atributos. Sin embargo, en el caso de BDOO, la AUM puede tener diferentes frecuencias de atributos para la misma transacción; por ejemplo, en el caso de atributos complejos, cuando el objeto hacia el cual dicho atributo tiene referencia puede ser accesado indirectamente por medio de este objeto complejo o directamente desde alguna otra transacción. Pues, el evaluador de particiones propuesto por Chakravarthy et al. [1] no puede ser aplicado sin modificaciones para el caso de BDOO.

Otro aspecto importante es que la mayoría de los algoritmos para la fragmentación vertical en BDDR usan la matriz de "afinidad" de los atributos como entrada. Dicha matriz no puede expresar la "afinidad" entre más de dos atributos, por consiguiente, una herramienta que usa la matriz de uso de atributos puede con más exactitud reflejar el comportamiento real del sistema. Además, algunos algoritmos existentes se declaran como los mejores y entregan las soluciones "óptimas" diferentes para los mismos datos de entrada. De este forma, el resultado de un algoritmo no puede ser comparado con los resultados de los otros, de aquí la necesidad de tener una herramienta para evaluar en qué medida los resultados son "óptimos". Sin embargo, esta herramienta fue desarrollada para BDDR después de que muchos de los algoritmos de fragmentación fueron presentados. Dicha herramienta para BDOO permite a los diseñadores evaluar los nuevos algoritmos de fragmentación conforme ellos se hacen disponibles.

4. DESARROLLO DE LA FUNCIÓN OBJETIVA

La meta de la fragmentación de los atributos es obtener el mínimo costo de procesamiento para un conjunto dado de transacciones con respecto al acceso a sus respectivos atributos. Es poco probable obtener un esquema de fragmentación ideal donde cada transacción accesa localmente solo los atributos que necesita y no necesitará accesar ninguno de los atributos de los lugares remotos. La función objetiva propuesta aquí trata de balancear el costo del acceso local y remoto para una transacción específica. De acuerdo con Chakravarthy et al. [1] el costo general en el ambiente distribuido que debemos minimizar consiste de dos elementos:

1. Costo de acceso de los atributos locales irrelevantes del sitio local, asumiendo que todos los datos requeridos por la transacción son localmente accesibles.

2. Costo de acceso de los atributos remotos relevantes requeridos por la transacción desde los sitios remotos. Chakravarthy et al. [1] han demostrado el comportamiento esperado de la función objetiva usando los componentes especificados anteriormente:

Individualmente, el costo de acceso de los atributos locales irrelevantes debería tener el valor máximo para un fragmento, donde están todos los atributos del sistema, y cero para la partición donde la cantidad de fragmentos es igual al número de atributos. Ocurre lo contrario con el costo del acceso a los atributos remotos relevantes con valor cero y máximo para estos dos extremos respectivamente. Además, como lo especifican los autores mencionados anteriormente, el costo local debería ser más sensible para los tamaños de particiones pequeñas, lo contrario del costo remoto donde éste debería ser más sensible para los fragmentos grandes.

Es evidente que el evaluador de particiones para BDOO debería tener el mismo comportamiento que se presentó antes para BDDR. Sin embargo, el costo del acceso de los atributos locales irrelevantes tiene un componente adicional. Debido que las frecuencias de uso de los atributos por la misma transacción pueden ser diferentes, se penaliza el acceso de algunos atributos locales relevantes cuando son accesados y no usados en la transacción. En este caso, estos atributos relevantes son tratados de igual manera que los atributos irrelevantes. Esta penalidad será incluida en el costo local.

**Bibliografía**

* <http://www.allbusiness.com/information/publishing-industries/399484-1.html>